CLAIMS

What is claimed is:

1. A method for implementing a software FIFO in a network processing engine, the method comprising:

receiving a request to write data to a FIFO;

determining whether the FIFO is full by comparing the value of a counting semaphore with a predefined maximum value; and

if the value of the counting semaphore is less than the predefined maximum value:

incrementing the counting semaphore; and

writing data to the FIFO.

2. The method of claim 1, further comprising:

receiving a request to read data from the FIFO;

reading the data from the FIFO; and

decrementing the counting semaphore.

3. The method of claim 2, in which the method is performed in the order recited.

4. The method of claim 1, further comprising:

receiving a request to read data from the FIFO;

determining whether the FIFO is empty by comparing the value of the counting semaphore with a predefined minimum value; and

if the value of the counting semaphore is greater than the predefined minimum value:

reading data from the FIFO; and

decrementing the counting semaphore.

5. The method of claim 4, in which at least said incrementing and decrementing are atomic.

6. The method of claim 1, further comprising:

if the value of the counting semaphore is not less than the predefined maximum value:

discarding the data that was to be written to the FIFO.

7. The method of claim 1, further comprising:

if the value of the counting semaphore is not less than the predefined maximum value:

blocking further execution of a process that made the request to write data to the FIFO until the value of the counting semaphore is less than the predefined maximum value.

8. The method of claim 1, in which the counting semaphore is implemented using special-purpose hardware.

9. The method of claim 8, in which the special-purpose hardware comprises a counter.

10. The method of claim 9, in which the special-purpose hardware further comprises at least one comparator for comparing an output of the counter with a predefined value and generating one or more signals based on the comparison.

11. A computer program product embodied on a computer readable medium, the computer program product comprising instructions which, when executed by a processor, are operable to perform actions comprising:

receiving a request to write data to a FIFO;

determining whether the FIFO is full by comparing the value of a counting semaphore with a predefined maximum value; and

if the value of the counting semaphore is less than the predefined maximum value:

incrementing the counting semaphore; and

writing data to the FIFO.

12. The computer program product of claim 11, further comprising instructions which, when executed by the processor, are operable to perform actions comprising:

receiving a request to read data from the FIFO;

reading the data from the FIFO; and

decrementing the counting semaphore.

13. A network processing engine comprising:

one or more coprocessors;

a memory;

a counting semaphore;

signal generation logic for signaling the status of a FIFO; and

computer code stored in said memory, which, when executed by one or more of said coprocessors, is operable to implement a FIFO using said counting semaphore and said signal generation logic.

14. A network processing engine as in claim 13, in which said signal generation logic is operable to generate a signal indicating whether or not the FIFO is full.

15. A network processing engine as in claim 13, in which said signal generation logic is operable to generate a signal indicating whether or not the FIFO is empty.

16. A network processing engine as in claim 13, in which said signal generation logic is operable to generate a signal indicating whether or not the FIFO contains more than a first predefined amount of data.

17. A network processing engine as in claim 16, in which said signal generation logic is further operable to generate a signal indicating whether or not the FIFO contains less than a second predefined amount of data.

18. A network processing engine as in claim 16, in which said first predefined amount comprises a number having the same number of bits as a maximum value of the counting semaphore.

19. A network processing engine as in claim 13, in which the signal generation logic comprises one or more comparators.

20. A network processing engine as in claim 13, in which the signal generation logic and the counting semaphore are implemented as part of the same circuit, the circuit comprising:

an adder/subtractor;

one or more comparators operatively connected to an output of the adder/subtractor; and

conditioning logic operatively connected to an output of the one or more comparators and an input of the adder/subtractor, the conditioning logic being operable to signal the adder/subtractor to increment or to decrement a value of the counting semaphore.

21. A network processing engine as in claim 13, in which the counting semaphore comprises a counter.

22. A network processing engine as in claim 21, in which the signal generation logic comprises at least one comparator for comparing an output of the counter with a predefined value and generating one or more signals based on the comparison.

23. A signaling method performed by a network processing engine, the method comprising:

maintaining a counting semaphore, the counting semaphore being operable to increment and decrement a count in an atomic fashion;

atomically incrementing the value of the count in response to a first action by a first process; and

taking at least one action in a second process based on the incremented value of the count.

24. The signaling method of claim 23, further comprising:

maintaining a plurality of signals derived from the count, the signals being modified in an atomic fashion.

25. The signaling method of claim 24, in which the count corresponds to an amount of data contained in a predefined portion of memory.

26. The signaling method of claim 24, further comprising:

atomically changing a state of a first of said signals in response to atomically incrementing the value of the count; and

taking at least one action based on the changed state of the first of said signals.

27. The signaling method of claim 25, in which the plurality of signals comprise an indication of whether the predefined portion of memory is full and an indication of whether the predefined portion of memory is empty.

28. The signaling method of claim 23, in which the counting semaphore is implemented using special-purpose hardware.

29. The signaling method of claim 28, in which the special-purpose hardware comprises a counter.

30. The signaling method of claim 29, in which the special-purpose hardware further comprises at least one comparator for comparing the count with a predefined value and generating one or more signals based on the comparison.